

Context Dependent Speech Recognition

Sebastian Andersson

**Linguistics and English Language
School of Philosophy, Psychology and Language Sciences
University of Edinburgh**

August 2006

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF
MASTER OF SCIENCE

© Copyright 2006
by
Sebastian Andersson

Declaration

I hereby declare that this thesis is of my own composition, and that it contains no material previously submitted for the award of any other degree. The work reported in this thesis has been executed by myself, except where due acknowledgement is made in the text.

Sebastian Andersson

Abstract

Poor speech recognition is a problem when developing spoken dialogue systems, but several studies have showed that speech recognition can be improved by post-processing of recognition output that use the dialogue context, acoustic properties of a user utterance and other available resources to train a statistical model to use as a filter between the speech recogniser and dialogue manager. In this thesis a corpus of logged interactions between users and a dialogue system was used to extract features from previous dialogue context, acoustics from the user utterance and n-best recognition hypotheses. The features were used to train maximum entropy models with different feature sets to rerank the n-best hypotheses. The models fail to some extent to predict intended labels but using the reranked output in effect means that 94.9% of the adequate hypotheses will be sent to the dialogue manager, a decrease in relative error over baseline with 44.6% showing that contextual reranking can improve speech recognition for dialogue systems. Future work involves developing the current feature sets and maxEnt models to better classify whether a hypothesis should be accepted or rejected by the dialogue system rather than rerank them.

Acknowledgements

I would like to thank my supervisor Oliver Lemon for showing interest and commitment throughout the project. Kalliroi Georgila for helping me with the recogniser problems. Ivan Meza-Ruiz for providing me with the parsability data. And all those who have helped me by answering my questions or just kept me company in the lab.

Contents

Declaration	ii
Abstract	iii
Acknowledgements	iv
Chapter 1 Context Sensitive ASR in Dialogue Systems	1
1.1 Modeling Dialogue Relevance	2
1.2 Background and Related Work	3
1.2.1 Evaluation Metrics	4
1.2.2 Features	5
1.2.3 Learning Methods	6
Maximum Entropy Modeling	6
1.2.4 The TownInfo Dialogue System	8
Chapter 2 Method	11
2.1 The TownInfo Dialogues	12
2.2 Generating N-best hypotheses	13
2.3 Labeling	15
2.3.1 Automatic Labeling	16
2.4 The TownInfo Corpus	17
2.5 Features - Types and Realisations	18

2.6	Implementation	21
2.6.1	The MaxEnt Toolkit	21
2.7	Combining Features and Final Testing	22
2.7.1	Test of Automatic Labeling	23
2.8	Baseline and Oracle	23
2.9	Qualitative Evaluation	24
Chapter 3	Results	27
3.1	Accept and Reject	28
3.2	Rerank	29
3.3	Automatic Labels	29
3.4	Error Analysis and Examples	30
Chapter 4	Discussion and Conclusions	33
4.1	Labeling	34
4.1.1	Automatic Labeling	34
4.2	Features	35
4.3	Method	37
4.3.1	Calculation Problems	38
4.4	Results	38
4.5	Future Work	39
4.6	Conclusions	40
Appendix A	Example of Results	41
References		60

CHAPTER 1

Context Sensitive ASR in Dialogue Systems

In an introductory text book to automatic speech recognition (Holmes and Holmes, 2001) the difference in performance between a human and a good speech recogniser is explained in part by that people benefit from *the linguistic redundancy of the speech signal* i.e. people know what they are talking about and can adapt their interpretation of the speakers utterance accordingly. A famous example of equally likely acoustic word sequences where the correct hypothesis has to be interpreted with regards to the dialogue discourse is: *'it's hard to wreck a nice beach'* or *'it's hard to recognise speech'*. But a speech recogniser is often used as a module in a natural language application e.g. a spoken dialogue system where the system also can be said to have certain expectations about the dialogue content; which utterances, keywords, grammatical structures and tasks that will lead the dialogue forward to task completion. In most cases there are only a limited number of tasks that the dialogue system can help the user to accomplish e.g. booking a movie ticket but not give directions to the nearest restaurant or give directions to the closest restaurant but not book movie tickets. In most cases the user also have a fairly good idea which subtasks (choose a film, a cinema, time, etc.) are involved in accomplishing the overall task (booking a movie ticket). So there should be a fair chance of enough 'common ground' to make it possibly for the system to guide the user to task completion although the user often has too high expectations of the system's abilities.

In a simple model of a spoken dialogue system the speech recogniser gives the most likely hypothesis about a user utterance to a dialogue manager that keeps track of the dialogue: which tasks have been completed, which task is currently performed and which tasks remains to be done, and provides the user with an appropriate output based on this knowledge. The process is then repeated, but whereas the dialogue manager keeps track of dialogue progression the speech recogniser works in isolation. Given the assumption that most user utterances are relevant to the dialogue then a recognition hypothesis that is relevant is more likely to match what the user really said and a hypothesis that is not relevant will also be rejected by the system. But the most likely hypothesis from a speech recogniser is based on the probability of the pronunciation and sequence of words and although this certifies the most likely word sequence according to the recogniser's internal model it doesn't guarantee that the hypothesis is correct, i.e. is what the user really said, or that it is a relevant utterance given the state of the dialogue. One way to incorporate a model of relevance in the dialogue system is to let the speech recogniser decode the n-best hypotheses and let a relevance measure be the basis for reranking the hypotheses. A domain specific recognition grammar might be good at ensuring that a recognition hypothesis is relevant, but just because there happens to be a hypothesis that seems relevant among the n-best the user might actually have said something irrelevant (perhaps talked to himself, a friend or coughed) and that can be equally important to detect.

1.1 Modeling Dialogue Relevance

There are a number of open research questions related to the above introduction of how to model dialogue relevance:

- How can speech recognition for dialogue systems gain from modeling contextual dependency of an utterance?
- Which contextual factors are important?
- How much dialogue history is necessary/sufficient?
- Does the representation of dialogue discourse have an effect?

- What is a good modeling framework to represent contextual dependency?

The study described in this thesis will address the questions above by using a corpus of user interactions with a dialogue system and extract features from several different contextual sources to build a statistical model to predict whether speech recognition hypotheses exhibit dialogue relevance. More specifically this thesis will investigate how post-processing of recognition n-best hypotheses can improve speech recognition for dialogue system purposes. Features are extracted from a Information State Update representation of dialogue discourse, acoustic properties of an utterance, linguistic properties of a recognition hypothesis and relational properties of the n-best list. The features are used to train Maximum Entropy models to rerank and classify the n-best lists according to their dialogue relevance.

1.2 Background and Related Work

Several studies have showed that dialogue systems benefit from post-processing of the speech recognition result. The data used in the studies mentioned in this section was user utterances and dialogue logs collected from interactions with different dialogue systems. The amount of available data ranges from a couple of hundred utterances (e.g. Gabsdil and Lemon 2004), to a couple of thousand (e.g. Litman et al. 2000) to over 30.000 utterances (Chotimongkol and Rudnicky, 2001). Different studies focused on different aspects e.g. detecting poor speech recognition through prosodic cues instead of recognition confidence scores (Litman et al., 2000), reranking or classification of n-best recognition hypotheses (Chotimongkol and Rudnicky 2001 and Gabsdil and Lemon 2004) or even integrating reranking of both recognition and parsing n-best lists in a dialogue system (Purver et al., 2006). The smallest common denominator being what several authors point out: poor speech recognition performance is a very important factor impairing the performance of dialogue systems (see e.g. Litman et al. 2000; Chotimongkol and Rudnicky 2001 or Gabsdil and Lemon 2004).

Litman et al. (2000) and Walker et al. (2000) worked on single top recognition hypotheses to detect poorly recognised speech and adapted dialogue system behaviour accordingly; how much information the system asked for in a turn and which confirmation strategy to use (Litman and Pan, 2000). Chotimongkol and Rudnicky (2001) tried to retrieve more correct recognition hypotheses by reranking n-best lists and thereby facilitating dialogue flow. Gabsdil and Lemon (2004) and Jonsson (2006) went one step further and tried to classify the n-best hypotheses with appropriate dialogue behaviour; whether to *accept*, *clarify*, *reject* or *ignore* a hypothesis (Gabsdil and Lemon, 2004) or similarly to choose 1 of 5 grounding strategies based on recognition classifications from *optimistic* - very confident about recognition output to *ignore* - definitely misrecognition (Jonsson, 2006). Purver et al. (2006) took on a more holistic dialogue system perspective in that they let the dialogue manager integrate several sources of information including recognition n-best lists to evaluate possible dialogue moves and select the overall highest scored move.

1.2.1 Evaluation Metrics

The most common measure for evaluating speech recognition performance is word error rate (WER), but Boros et al. (1996) argue convincingly from examples that *concept* error rate (CER) that takes into account semantic aspects of the difference between a transcript and a recogniser hypothesis is a more adequate evaluation metric than WER for speech recognition in a dialogue context. ‘Semantic aspects’ was based on the systems point of view i.e. which content words and syntactic constructs the system can interpret as valid dialogue acts and is calculated in the same way as WER but replacing words with semantic units (Boros et al., 1996). WER and CER was also the evaluation metrics in Litman et al. (2000) and Chotimongkol and Rudnicky (2001) with the CER adapted to their specific dialogue system. Jonsson (2006) used a similar evaluation metric to CER but used no partial matches: If a hypothesis correspond to the same dialogue move as the transcript they exhibit 100% dialogue move accuracy otherwise 0%. Gabsdil and Lemon (2004) evaluated their machine learning experiment with prediction accuracy and weighted F-score.

1.2.2 Features

The features used in the studies described in this chapter are overlapping to a large degree. Acoustic features from a user utterance was used in several of the studies (e.g. Walker et al. 2000; Chotimongkol and Rudnicky 2001 or Gabsdil and Lemon 2004) and Litman et al. (2000) showed that there was significant difference in acoustic properties (e.g. energy (RMS) and duration) between correctly recognised and misrecognised utterances.

Features from the dialogue system's representation of context was also used to a large extent and to a varied degree. In particular representations of previous system question (and the task it represent) and whether a hypothesis is a valid answer to that question was used (e.g. Chotimongkol and Rudnicky 2001; Gabsdil and Lemon 2004). And also which 'slots' in the dialogue representation would be filled by the hypothesis (Jonsson, 2006; Purver et al., 2006). Jonsson (2006) used the most dialogue context features, e.g. all previous dialogue moves, previously agreed actions and how many recogniser rejections the dialogue history contained. Another interesting dialogue feature was whether the task that a hypothesis represented would contradict a task that is already going on (Gabsdil and Lemon, 2004).

Recogniser confidence scores for hypotheses and/or individual words was also commonly used (Litman et al. 2000; Gabsdil and Lemon 2004; Jonsson 2006 and Purver et al. 2006). The actual representation of the features might differ a little e.g. Gabsdil and Lemon used only the minimum word confidence score whereas Jonsson used confidence score for every word.

Walker et al. (2000) and Litman et al. (2000) use the recognition output per se, i.e. the words and hypothesis as a feature. But even though it proved to be the strongest feature Litman et al. raised a concern that it might be too domain and recogniser dependent.

1.2.3 Learning Methods

A number of different machine learning or statistical modeling techniques have been used in the studies mentioned under the *Background and Related Work* section. The learners took a set of features and a set of outcome classes aligned with a recognition hypothesis as training data and builds a model to predict the outcome class for a previously unseen hypothesis.

Chotimongkol and Rudnicky (2001) used a linear regression model built from their features. Litman et al. (2000) used RIPPER (Cohen, 1996) that builds a model expressed as an ordered set of if-then rules. RIPPER was also used in Gabsdil and Lemon (2004) but was discarded because a memory based learner, TiMBL (Daelemans et al., 2002) performed better. A memory based learner works by storing training instances in memory and extrapolating from the instances to predict the outcome of test data. The idea behind memory based learning is to adopt an approach to learning that is supposedly cognitively plausible in that it draws conclusion directly based on previous experiences instead of from an abstracted set of rules (Daelemans et al., 2002). TiMBL was also used in Jonsson (2006) with a satisfactory result. Purver et al. (2006) used manually set threshold values in their experiment but plan to use machine learning in the future.

Maximum Entropy Modeling

In the study described in this thesis Maximum Entropy modeling was used. It is a well-established supervised machine learning framework that has been shown to perform well on various tasks in natural language processing such as part-of-speech tagging (Ratnaparkhi, 1996), machine translation (Och and Ney, 2002) and named entity recognition (NER) (Tjong Kim Sang and De Meulder, 2003). In relation to the study described in this thesis Maximum Entropy modeling has also been used to predict an appropriate parse from n-best parses for an utterance in a spoken dialogue system (Meza-Ruiz and Lemon, 2005) and with ASR n-best hypotheses for improving NER on Chinese speech (Zhai et al., 2004).

The following paragraphs and formula notation concerning Maximum Entropy modeling is based on the description in Berger et al. (1996) unless stated otherwise. Berger et al. divides the maximum entropy modeling problem into two parts: finding facts about the data and incorporate the facts into a model. Facts are represented as feature functions (or features for short) and the intuition behind incorporating the facts into a model can be stated as: “[...] model all that is known and assume nothing about that which is unknown”(Berger et al., 1996, p. 3).

The feature functions output a value given a context x (e.g. the word *Chinese* in an ASR hypothesis) and an outcome y (e.g. An ‘*accept*’ label) in a data sample. In the study described in this paper and the description of Maximum Entropy modeling in Berger et al. (1996) only binary feature functions are used:

$$f(x, y) = \begin{cases} 1 & \text{if } y = \textit{Chinese} \text{ and } x = \textit{accept}, \\ 0 & \text{otherwise} \end{cases}$$

(Example adapted from Berger et al. 1996)

The expected value of f given the observed probability of (x, y) in the data sample is given by:

$$\tilde{p}(f) \equiv \sum_{x,y} \tilde{p}(x, y) f(x, y)$$

To incorporate this statistics in the model the model’s expectation $p(f)$ is calculated:

$$p(f) \equiv \sum_{x,y} \tilde{p}(x) p(y|x) f(x, y)$$

In practice $p(f)$ is too complex to be computed directly, because of the large number of possible contexts, (x) and is approximated (Ratnaparkhi, 1996).

To constrain the model's expectation $p(f)$ to represent the statistics that has been observed in the data $\tilde{p}(f)$ a requirement is set:

$$p(f) = \tilde{p}(f)$$

The log-likelihood of the model is given by:

$$L_{\tilde{p}}(p) \equiv \sum_{x,y} \tilde{p}(x,y) \log p(x|y)$$

The entropy of the distribution $p(y|x)$ is given by:

$$H(p) \equiv - \sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x)$$

The entropy is then maximised with maximum likelihood estimation. The result is a model that given a set of features calculates a set of weights ($p(y|x)$) that constrains the features and maximises the log-likelihood of the model. The model can then be applied to predict the outcome given a context.

The weights can for most real problems not be calculated analytically and has to be found with numerical methods (Berger et al., 1996). Malouf (2002) showed that the L-BFGS algorithm performed much better on natural language processing tasks than the more commonly used Generative Iterative Scaling (GIS) algorithm because of the vast number of parameters used in NLP.

1.2.4 The TownInfo Dialogue System

The dialogues used in the study described in this thesis was collected from user (both native and non-native) interactions with the TownInfo dialogue system (Lemon, Georgila and Henderson, 2006) developed within the TALK project (<http://www.talk-project.org>) which is an experimental system where presumptive users are people travelling by car to a city or are in the city and interacts with the system through natural speech to book a

room/table at a hotel/restaurant or find a bar. All tasks involved subtasks of preferred choice regarding price range, location and type of facility (Lemon, Georgila, Henderson and Stuttle, 2006). The dialogue system was implemented in the Open Agent Architecture (OOA) (Cheyer and Martin, 2001) with the main components being a dialogue manager, a dialogue policy reinforcement learner, a speech recogniser and a speech synthesiser. The components of interest for this thesis is the dialogue manager and the speech recogniser.

The dialogue manager, DIPPER (Bos et al., 2003), is an Information State Update (ISU) approach to dialogue management that was specifically developed to handle spoken input/output and integrates several communicating software agents to keep track of dialogue progression and let the agents communicate about what action should be taken based on previous and current state of the dialogue and save the information in logs with filled slots e.g. user utterances, recogniser confidence, system output, previous tasks performed, current task, speech acts etc. The architecture offers for e.g. recogniser confidence scores to guide the dialogue manager's clarification and confirmation strategies that is then given to the speech synthesiser to realise (Lemon, Georgila, Henderson and Stuttle, 2006).

The speech recogniser was built with the ATK toolkit (Young, 2004). The recogniser uses a statistical (n-gram) language model that combines in-domain data with general purpose data instead of a domain specific recognition grammar network. The advantage of this type of statistical language model is that new language models can relatively easily be adapted to modifications of the dialogue system without impairing the WER of the recogniser or having to build precise recognition grammars (Weilhammer et al., 2006).

CHAPTER 2

Method

The data consisted of 191 dialogue logs collected from user interactions with a spoken dialogue system containing a total of 2904 user utterances saved in sound files. Each utterance was transcribed and n-best recognition hypotheses were generated off-line resulting in a total of 15819 hypotheses. 67 utterances did not generate any hypotheses at all reducing the total number of usable utterances to 2837. Each hypothesis was labeled with a 'B'(close enough to transcript) or 'N' (not close enough to transcript). Features were extracted from dialogue logs, sound files and n-best lists and used to train and test Maximum Entropy models. The final testing of the Maximum Entropy models was done with 10-fold cross-validations. The motivation for not testing on a test-set of previously unseen data was that no truly unseen data was available since the author of this thesis transcribed and labeled all data. The final testing was done on two different feature sets: a larger set with a combination of acoustic, dialogue discourse and n-best features with a total of 9 features and a smaller set with four features: hypothesis, preceding system question, every word in the hypothesis and all previous ASR top-hypothesis. The first two features (hypothesis and preceding system question) were also included in the larger feature set.

System: *how can i help you?*
User: *i need to find a hotel room*
System: *did you say hotel?*
User: *yes*
System: *do you want something central?*
User: *yes please*
System: *do you need a single or double room?*
User: [empty]
System: *do you need a single or double room?*
User: [empty]
System: *do you need a single or double room?*
User: *single*
System: *would you like something cheap, expensive,
or reasonably priced?*
User: *i dont mind*
System: *now i am going to search for
a single central room. Please wait a moment while
i search for suitable hotel options.*
(system presents options)
*are you interested in any hotel option number or
would you like me to keep going?*
User: *number three*
(system repeats option number three)
System: *are you happy with that choice?*
User: *yes*
System: *would you like to search some more?*
User: *no*
System: *goodbye.. and thanks for using the
edinburgh cambridge dialogue system.*

Figure 2.1: Example of a completed dialogue between the system and a user.

2.1 The TownInfo Dialogues

The dialogue logs and utterances used in this study was collected from user interactions with the TownInfo dialogue system (see section 1.2.4). The users use natural speech to interact with the system and perform tasks like booking a hotel room or a table at a restaurant, an example dialogue is shown in figure 2.1.

The users were both native and non-native English speakers, the non-natives sometimes had clear foreign accents and some grammatical errors but at no time was it difficult for the (non-native) author of this thesis to understand them. The dialogue logs were structured according to the Information State Update (ISU) approach (see figure 2.2).

Approximately 10 dialogues were not included in the final corpus of dialogue logs due to that they were recordings of the experimenters trying out the interaction before collecting data from experiment participants. Since the dialogues are already collected in an experimental environment it was considered unnecessary to introduce even more artificiality and perhaps skew the data distribution. However one dialogue full of *'stop'* and *'restart'* commands was included to allow for more training data for this type of utterance that was not very common among the dialogues collected from the experiment participants.

2.2 Generating N-best hypotheses

When the dialogue logs were generated the speech recogniser was operating on direct audio input and only gave the top hypothesis as input to the dialogue manager. The utterance itself was saved in a wav-file, therefore n-best recognition hypotheses were decoded off-line from the utterance wav-file with AVite from the ATK toolkit (Young, 2004) using the same configuration settings; dictionary, n-gram language model, hmm models and beam widths as during the collection of dialogue logs; the only differences in configuration settings being that n-best output was generated and that recognition decoding was performed on wav-files instead of direct audio. This resulted in the top-hypothesis often not being the same as in the originally collected dialogues and sometimes the original top hypothesis was not present at all in the off-line collected n-best lists, something that apparently can happen due to delayed silence detection, (Georgila, K., personal communication). But it was not deemed to have a major impact on the result of this study because only previous utterances in the dialogue logs was used to extract features for the current utterance and for previous utterances the original top-hypothesis and the dialogue log created was available.

A maximum of seven n-best hypotheses was generated for each of the 2904 utterances. The number of generated hypotheses was sometimes below seven (or even zero) because of the beam widths in the configuration settings, although they make recognition faster sometimes they prevent hypotheses from being decoded. The number 7 was somewhat

```

STATE 7
DIALOGUE LEVEL
TurnNumber: 3
Speaker: user
DialogueActType: user
ConvDomain: about_task
SpeechAct: [provide_info]
AsrInput: <s> yes please </s>
TransInput: yes please
TASK LEVEL
Task: [hotel_location]
FilledSlot: [hotel_location]
FilledSlotValue: [central]
LOW LEVEL
AudioFileName:
tests-003-2006-03-03_13-39-21.wav
ConfidenceScore: 0.47
HISTORY LEVEL
PreviouslyFilledSlots:
,[null],[top_level_trip],[null],[hotel_location]
PreviouslyFilledSlotsValues:
,[null],[hotel],[],[central]
SpeechActsHist:
opening_closing,request_info,[provide_info],
explicit_confirm,[yes_answer],request_info,
,[provide_info]
TasksHist:
meta_greeting_goodbye,top_level_trip,[top_level_trip],
top_level_trip,[top_level_trip],hotel_location,
,[hotel_location]
FilledSlotsHist: [top_level_trip],[],[hotel_location]
FilledSlotsValuesHist: [hotel],[],[central]

```

Figure 2.2: Part of a dialogue log showing the information state fields and values (<Field>:<value>) for the user utterance ‘yes please’ as an answer to the system’s question about hotel location *Would you like something central?*.

arbitrarily chosen but the decision was based on the presence of correct hypotheses (i.e. matching transcript) further down the n-best list. As mentioned in the introduction of this chapter 67 utterances did not generate any hypotheses at all resulting in 2837 utterances and a total of 15819 hypotheses.

2.3 Labeling

Each of the n-best hypotheses was hand labeled with a 'B' or 'N' label. There are two ways to interpret the labels: One is to consider the 'B' label as a hypothesis the dialogue system should *accept* and the 'N' label as a hypothesis the dialogue system should *reject*. The other one is to see the labels as reranking markers where a 'B' outranks an 'N'. Why these two interpretations are both fruitful will be discussed in chapter 4.

A 'B' label was assigned when a hypothesis was considered semantically *close enough* to the transcript of the corresponding utterance, and an 'N' label was assigned when a hypothesis was considered to be too far away from the semantics of the corresponding transcript. The decision whether a hypothesis was semantically *close enough* to the transcript was based on the semantics of the dialogue system where the presence of correct (i.e. matching transcript) keywords makes it a valid dialogue move. To this semantic base-rule some additional guidelines were added: The hypothesis should not contain negations or keywords that are not present in the transcript and that might confuse the dialogue system's interpretation of the hypothesis (e.g. words related to other tasks than the one at hand). The second guideline was: If the user says something that the system can't handle then the hypothesis has to be semantically close enough from a human perspective (e.g. '*could you repeat the hotel name*' is semantically close enough to '*can you repeat that hotel name, please*' but neither one is a valid dialogue move in the current system). The third guideline was that if the user utterance was not directed to the system (e.g. directed to the experimenter, self talk or laughter) all hypotheses are labeled 'N'. The fourth guideline was that if none of the keywords in the transcript was in the hypothesis then the hypothesis was labeled 'N'. The fifth and last guideline was that a hypothesis was labeled 'N' if another hypothesis in the n-best list contained more of the

Transcript: indian restaurant YEAH RESTAURANT 'N' IT YEAH RESTAURANT 'N' INDIAN RESTAURANT 'B'	Transcript: drink THANK YOU 'N' DON'T CARE 'N' I DON'T CARE 'N'
Transcript: find me bar I NEED A BAR 'B' I MEAN A BAR 'B' CARTE ME A BAR 'N'	Transcript: number two NUMBER TWO 'B' NOT A TWO 'N' NUMBER TO 'N'

Figure 2.3: Four examples of the transcript and labeled n-best list.

keywords in the transcript, e.g. *'yeah restaurant'* was labeled *'N'* when the transcript and another hypothesis was *'indian restaurant'*. Figure 2.3 shows four examples of labeled n-best lists (more examples can be found in appendix A).

2.3.1 Automatic Labeling

In addition to the hand labels an attempt was made to develop an automatic label system based on the Levenshtein distance between a hypothesis and its corresponding transcript. The reason for developing an automatic label system was that the labeling described in section 2.3 was, like almost all hand labeling, time consuming and suffer from labeling errors and inconsistencies. The attempt was meant to indicate the effort that would be needed to develop a fully automatic labeling system based on this approach.

In a first simple automatic labeling system a hypothesis was labeled *'B'* if it was the closest one to the transcript or if it was one distance metric away from the closest hypothesis. A hypothesis was labeled *'N'* if it didn't match the above condition or if the Levenshtein distance was larger than the number of words in the hypothesis. This will prevent one word hypotheses with a one word corresponding transcript to always be labeled *'B'* and it will prevent hypotheses that are too *'scrambled'* from being labeled with *'B'*. In addition to this simple system a small set of words that were considered *'meaningless'* was removed from the hypothesis before calculating the Levenshtein distance and some words that were considered synonyms were collapsed into one word. The meaningless words were: *'ah'*, *'eh'*, *'er'*, *'oh'*, *uh'* and *'um'*. The words that were considered synonyms

Transcript: indian restaurant YEAH RESTAURANT 'B' IT YEAH RESTAURANT 'N' INDIAN RESTAURANT 'B'	Transcript: drink THANK YOU 'N' DON'T CARE 'N' I DON'T CARE 'N'
Transcript: find me bar I NEED A BAR 'B' I MEAN A BAR 'B' CARTE ME A BAR 'B'	Transcript: number two NUMBER TWO 'B' NOT A TWO 'N' NUMBER TO 'B'

Figure 2.4: Four examples of the transcript and automatically labeled n-best list.

and collapsed into one word were: 'ok'/'okay', 'the'/'a'/'an', 'yes'/'yeah'/'yep', 'no'/'nope' and 'I am'/'I'm'. The 'meaningless' words and synonyms are by no means exhaustive or unquestionable and other potential synonyms could probably be found by analysing decoder output. Figure 2.4 shows four examples of (advanced) automatically labeled n-best lists (same hypotheses as in figure 2.3)

2.4 The TownInfo Corpus

All user utterances were transcribed and the transcription was added to an appropriate slot in the dialogue ISU logs. The transcripts tried to capture the intended word rather than the actual pronunciation of a word, e.g. foreign accents were ignored and a pronunciation of e.g. [hotel] as [xotɛl] was transcribed as 'hotel'. Square brackets were used to surround transcripts of speech not directed to the system; transcribed as [crosstalk], wav-files that didn't contain any audible words; transcribed as [], and some other places where it was considered useful to understand why the recognition hypothesis looked the way it did e.g. *Transcript: yes [laughter]* and corresponding *Hypothesis: yes and where* *er*.

The hand labeled n-best lists were also added to the ISU logs of the dialogue. The dialogue logs were collected in two different experiment sessions and also differed a little in which information was included in the logs e.g. file path to wav-file was missing in one set of logs, since that was considered useful information they were added.

In the end the corpus used for this study consisted of 191 dialogue logs, including ISU fields and values, transcripts of 2904 utterances, labeled n-best lists and file paths to wav-file locations.

2.5 Features - Types and Realisations

Features was extracted from the dialogue logs, sound files and n-best lists based firstly on the author's informed believes from exposure to the data set, secondly on what features have been used in previous studies of similar type (see chapter 1) and thirdly on what mistakes the classifier made.

Several different types of features and representations of features were implemented and tested in different configurations with a maximum of four cross-validations on four sets of the data that were used as development sets to see whether the features might have a positive impact on the result. At this stage some features and feature representations were considered not to have any positive impact on the result and were only occasionally added later on to confirm this belief. Among these discarded features were e.g. previously filled slot values (see figure 2.2), preceding system speech act and all attempts to include features from the content (words or whole hypothesis) of other hypotheses in the n-best list than the current one.

The features can be broadly classified into three groups depending on whether they were extracted from dialogue logs (discourse), sound files (acoustic) or n-best lists (hypothesis):

1. **Discourse Features:** preceding system question, all previous asrInput, current task.
2. **Acoustic Features:** min/max/mean amplitude, duration.
3. **Hypothesis Features:** full hypothesis, every word in hypothesis, bigrams in hypothesis (plus relative frequency in training set), length in words, parsability, hypothesis rank in n-best list, number of hypotheses in n-best list.

The discourse features were all available from the dialogue logs and represented as literal strings:

Preceding system question is the dialogue system's question that was just asked and that the user presumably answers. Intuitively this seems to be a good feature for classifying whether a hypothesis about the answer is a valid answer.

Current task is a tag in the system log stating which task is currently handled in the dialogue e.g. 'hotel_location'.

All previous asrInput, or as they are referred to in the first paragraph *all previous ASR top-hypothesis*, is the recogniser input from previous user utterances and is the hypothesis that had an effect on the content of the dialogue logs. There were two reasons for including this feature: the same user seemed often to re-use the same phrasing e.g. *I'd like a hotel room/beer/restaurant* or consistently answered *yes I did*. The other reason is less fortunate and was due to the dialogue system's habit of repeating the same question more than once, even though it seemed from the dialogue log to understand the answer, and the user very often gave the same answer again.

The acoustic features were included mostly to try and classify *rejects* better and because they were used in several other studies of similar kind with satisfactory result. The acoustic features were extracted through the Unix sox utility:

The *min*, *max* and *mean amplitudes* were represented as strings of 'type: integer value', e.g. 'Max: 11'.

Duration of the sound files were categorised into 'very short' - shorter than 1.2s, 'short' - shorter than 1.5s, 'medium' - shorter than 15s, and 'long' - over 15s. The categorisation of the duration was based on inspection of the duration of *yes/no*-answers vs. hypotheses transcribed as '[]' *empty* resulting in the 'very short', 'short' and 'medium' boundaries, and duration of files containing crosstalk which resulted in the 'long' classification.

The hypothesis features relates to the content of the hypothesis and the relation to other hypotheses in the n-best list. The content of the hypothesis was included because together with the question asked it was considered to be the strongest reason for which label it got.

Hypothesis: The full hypothesis represented as a string.

Every word in a hypothesis was included to capture that not only the full hypothesis matter but also which keywords are present.

Bigrams in a hypothesis together with its relative frequency in the training set ('very low' = 1, 'low' < 5, 'medium' < 100, 'high' < 500 and 'very high' > 500). The category levels were set somewhat arbitrarily but based on the frequencies occurring in training data. The bigrams and their frequencies were included to reduce the impact of single words with high frequencies but low importance such as determiners and prepositions.

Length, i.e. number of words, was used because it was easy to implement and it has been considered a relevant feature in other studies.

Hypothesis rank was considered as a useful feature because most top-hypothesis was good hypothesis, i.e. 'B' labeled.

Number of hypotheses in n-best list was included because fewer hypotheses seemed to indicate that at least one was correct, in particular when there was only one or two hypotheses in the n-best list.

Parsability was included since most transcripts were considered fairly grammatical whereas many bad hypotheses often seemed to be ungrammatical and parsability might therefore improve both correct *accepts* and correct *rejects*. The information whether a feature was parsable or not was provided by Meza-Ruiz, I. based on a robust version of the broad coverage English Resource Grammar (ERG): <http://www.delphin.net/erg/>.

2.6 Implementation

The software modules that extracted features from the corpus and trained and tested the MaxEnt models was implemented in Python (www.python.org) because of the available MaxEnt toolkit (see section 2.6.1) and the author's acceptable skills in Python programming. In addition to the main implementation described below programs were also written to facilitate labeling, matching labeled n-best lists to its place in a dialogue, building of the corpus and to do automatic labeling. The acoustic and parsability features were extracted for every utterance and hypothesis respectively on beforehand and put into Python dictionaries accessible by the main program.

The main program consisted of four modules: A feature extractor, a maxEnt trainer, a maxEnt tester and a scoring module. The feature extractor extracted features from a chosen subset of the dialogue corpus and from the acoustic and parsability dictionaries and aligned them with the appropriate hypothesis. The maxEnt trainer trained a maxEnt model with a chosen set of features from a chosen set of dialogues. The maxEnt tester extracted features and tested the trained maxEnt model on a chosen subset of the features and corpus. The maxEnt tester also wrote relevant parts of the test data (transcript, hypotheses, some features and predicted and hand labels) to file to enable qualitative inspection of the results (see Appendix A). The scoring module calculated the numbers used to calculate the result.

2.6.1 *The MaxEnt Toolkit*

In this study Zhang Le's implementation of Maximum Entropy modeling (Le, 2004) was used because it is freely available (<http://homepages.inf.ed.ac.uk/s0450736/maxent.html>), fast and easy to use. The toolkit was written in C++ but comes with both a Python binding and a Unix command line utility. The command line utility offers more customised settings but the Python binding was considered good enough for this study and it was also the version that the author was familiar with.

In this study MaxEnt was used with the L-BFGS algorithm for parameter estimation (default value). The number of iterations were set manually and iterations between 25-150 in steps of 25 were tested to try and find a reasonable number of iterations for maximum performance on the test data. During training it is possible to see progress of prediction accuracy on the training data which when compared to test data offers some insights to possible 'overfitting' (in the command line utility a 'held-out' test set can be tested alongside training to see where MaxEnt gives the best result for this held-out set).

Features or feature functions are represented as literal strings with a possibility to set feature values. In this study only binary feature functions were used. Features that only occur once in training are cut-off by MaxEnt.

2.7 Combining Features and Final Testing

From initial tests of features in solitude and combinations with four times cross-validations it was clear that the words in the hypotheses was a very strong feature, followed by the hypothesis as a whole and previous system question. These features together with all previous *asrInput* were among the first to be implemented and tested and because they gave good performance when tested additional features were added to this small feature set. But even though some features didn't impair the result too much no features really improved it either.

Because the result seemingly couldn't be improved on the small feature set: *hypothesis*, *every word* in the hypothesis, *preceding system* question and *all previous asrInput*, it was tested in a final 10-fold cross-validation where approximately 90% of the data was used for training a MaxEnt model which was tested on the remaining 10% of the data and repeated ten times. The implementation structure only permitted training and testing on full dialogues, therefore the number of utterances used for training and test differs to some extent between the cross-validations. The result from this testing was then used as a guideline performance when trying to build up another feature set from 'scratch' (if it didn't reach or come close to this performance it would be discarded) The features

every word (considered too strong) and *previous asrInput* (considered opaque in terms of qualitative impact) was left out and one set of the data that was considered ‘hard’ was chosen as a development set and features was added one by one to the two features *full hypothesis* and *previous system question* and if a feature improved the result by a satisfactory amount it was kept in. The procedure was iterated until all features was tested and features that didn’t have an effect in one test run was re-tested every time a new feature was added. When all other features had been tested the *every word*, *asrInput* and *bigram + relative freq.* was tried, and only the bigram feature had a positive impact. This new feature set was then tested in the same 10-fold cross-validation set-up as the first feature set.

The final testing was hence made on two feature sets:

1. **The Small Feature Set:** hypothesis, previous system question, every word in hypothesis, all previous asrInput.
2. **The Big Feature Set:** hypothesis, previous system question, min amplitude, number of hypothesis, parsability, current task, hypothesis rank, length of hypothesis, bigram + relative freq.

2.7.1 Test of Automatic Labeling

The big feature set was also trained on automatic labels and tested on hand labels in the same 10-fold cross-validation set-up as the other experiments. The training was done with the two versions of automatic labeling described in section 2.3.1. Only the big feature set was trained and tested in this way because of lack of time and informal testing indicated that it had greater potential.

2.8 Baseline and Oracle

The baseline was calculated from using only the top-hypothesis of the n-best lists and calculating how many times the top is labeled ‘B’ or ‘N’ which would represent how many times the dialogue system *should accept* or *reject* a hypothesis. If the actual top

hypotheses were used in the dialogue system the behaviour would be slightly different because of false positives, e.g. the top hypothesis is 'no' which is a valid act in the dialogue system but the transcript is 'yeah'.

The oracle was calculated by counting how many times there was a 'B' labeled hypothesis anywhere in the n-best list, thereby increasing the number of *accepts*, and showing the upper limit of optimal performance with this recogniser.

The evaluation is not made on overall accuracy and precision of predicted labels but whether the predicted 'top' ranked hypothesis is correct. A correct-B was defined as: The highest ranked hypothesis with a predicted 'B' corresponds to a hand labeled 'B'. A false-B was defined as: The highest ranked hypothesis with a predicted 'B' corresponds to a hand labeled 'N'. A false reject was defined as: Every hypothesis in the n-best list was predicted 'N', but one of the hypothesis was hand labeled with a 'B'. A correct reject is defined as: Every hypothesis in the n-best list was predicted and hand labeled with 'N'.

There is no baseline score for *rejects* since the main purpose of this study was to find the good hypothesis, and *rejects* were considered more as a possible future extension of the work.

2.9 Qualitative Evaluation

In addition to the quantitative testing a qualitative evaluation was performed on two data samples from the final test runs (in total 31 dialogues containing 497 utterances). The samples were chosen on the basis of the contrast in performance between the two features sets, but because of lack of time error classification was only done for the big feature set.

Informal qualitative analysis was performed throughout the developing phase and the final qualitative analysis was mainly done to get a more structured perspective of which

phenomena are problematic to the current feature sets and to guide future feature selection.

CHAPTER 3

Results

Like it was written in chapter 2 but repeated here for convenience, the final testing was made on two feature sets:

1. **The Small Feature Set:** hypothesis, previous system question, every word in hypothesis, all previous asrInput.
2. **The Big Feature Set:** hypothesis, previous system question, min amplitude, number of hypothesis, parsability, current task, hypothesis rank, length of hypothesis, bigram + relative freq.

The calculated baseline and oracle score for 'B' labeled hypothesis was 1890 and 2081 hypothesis respectively out of the total number of 2837 decoded utterances which make the baseline 90.8% of the oracle score. The oracle score subtracted from the total number of utterances gives the total number of rejects: 756 or 26.6% of the total corpus.

All results from the MaxEnt models reported in this chapter are achieved through training with 100 iterations, since this was the amount of training that proved to be best for both feature sets.

3.1 Accept and Reject

Accepts and rejects are measured with precision, recall and F1 score. The raw number of predicted accepts for both feature sets are shown in tables 3.1 and 3.3. The raw numbers show that summed over all 10 cross-validations neither feature set beat the baseline. The big feature set beat the baseline in half of the test batches whereas the small feature set only reached the baseline twice. Tables 3.2 and 3.4 show precision, recall and F1 on predicting *accepts* or 'B' labels for the small and big feature set.

baseline-B	193	189	173	208	157	179	221	235	228	107	1890
oracle-B	204	218	197	221	171	196	246	255	254	119	2081
Results:											
correct-B	186	177	180	205	151	177	221	230	214	97	1838
false-B	19	39	72	66	47	18	41	39	58	23	422

Table 3.1: 10-fold cross validation results for predicting 'B' labels for the small feature set.

Precision	Recall	F1
81.3%	88.3%	84.7%

Table 3.2: Precision, recall and F1 score from 10-fold cross validations for predicting 'B' labels for the small feature set.

baseline-B	193	189	173	208	157	179	221	235	228	107	1890
oracle-B	204	218	197	221	171	196	246	255	254	119	2081
Results:											
correct-B	192	191	174	212	153	177	217	228	229	108	1881
false-B	17	37	48	22	27	16	35	21	45	24	292

Table 3.3: 10-fold cross validation results for predicting 'B' labels for the big feature set.

Precision	Recall	F1
86.6%	90.4%	88.5%

Table 3.4: Precision, recall and F1 score from 10-fold cross validations for predicting 'B' labels for the big feature set.

There were a total of 756 n-best lists that were labeled as rejects (all hypotheses in the list labeled 'N') in the data, calculated by subtracting the oracle-B from the total number of utterances. The big feature set correctly rejected 533 out of 756, but falsily rejected 131 n-best lists. The small feature set correctly rejected 466 n-best lists, but falsily rejected 149 n-best lists. Precision, recall and F1 score for both feature sets are shown in table 3.5.

Feature set	Precision	Recall	F1
Small	75.8%	61.6%	68.0%
Big	80.3%	70.5%	75.1%

Table 3.5: Precision, recall and F1 for both feature sets (Big and Small) on predicting rejects.

3.2 Rerank

Using the 'B' and 'N' labels as reranking markers where a 'B' outranks an 'N' would mean that a score would be calculated by adding the correct-B to the false rejects that have the top hypothesis labeled with 'B':

$$\text{correct-B} + \text{false rejects with a top hand labeled 'B'} = \text{sum correctly ranked}$$

For the big feature set the total number of false rejects with a hand labeled 'B' as top hypothesis was 94, and for the small feature set it was 121. Adding these numbers to the sum of correct-B in table 3.1 and 3.3 and divide with the oracle score give the following percentage of correctly ranked hypotheses:

Baseline-B: 90.8% of the oracle score

Big feature set: 94.9% of the oracle score.

Small feature set: 94.1% of the oracle score.

The total baseline error (compared to the oracle) is then 9.2% and the error for the big feature set is 5.1%, an absolute improvement of 4.1%, but also a relative improvement of error of 44.5%. For the small feature set corresponding figures are 3.3% absolute improvement and 35.9% relative improvement.

3.3 Automatic Labels

The result from training with the big feature set on automatic labels and testing on hand labels is shown in figure 3.6 for the same 10-fold cross-validation set-up as in previous tests.

Baseline-B	SimpleAuto-B	AdvancedAuto-B
193	195	194
189	167	181
173	163	167
208	206	206
157	153	154
179	184	183
221	216	219
235	233	232
228	225	218
107	109	112
Sum:1890	1851	1866

Table 3.6: Results from 'B' predictions when training with the big feature set and simple and advanced automatic labels and testing on hand labels.

The number of false-B for the advanced automatic labels were 618 and for the simple automatic labels it were 513 which gives the precision, recall and F1 score shown in table 3.7.

Labels	Precision	Recall	F1
Simple	78.3%	88.9%	83.3%
Advanced	75.1%	89.7%	81.7%

Table 3.7: Precision, recall and F1 score from 10-fold cross validations for predicting emph'B' labels for the big feature set trained on advanced automatic labels.

3.4 Error Analysis and Examples

As can be seen from the numbers in the previous section MaxEnt most of the time predicts correct labels and qualitative analysis confirms the numbers. Figure 3.1 show some cases of good predictions for the big feature set (more examples can be found in Appendix A).

Two batches of cross validations (in total 497 n-best lists) of the big feature set were manually classified with types of errors the MaxEnt model made. In total 79 errors were found and classified. 8 occurrences of mislabeling, i.e. labels where the labeling guidelines clearly had been violated, was spotted during this process. Figure 3.2 show some examples where the MaxEnt system made wrong predictions (more examples can be found in Appendix A).

Transcript	Hypotheses	human/maxEnt labels
something expensive	SOMETHING EXPENSIVE	B/B
	SOMETHING INEXPENSIVE	N/N
	TELL ME EXPENSIVE	N/B
	SOMETHING A EXPENSIVE	B/B
[laughter]	HOW ABOUT A THEY'RE OR	N/N
	YOU ARE THERE THERE ARE	N/N
	HOW ABOUT A THERE ARE	N/N
	ARE THERE THERE ARE	N/N
i need a hotel	I NEED A CARTE NOW	N/N
	I NEED A HOTEL	B/B
	I NEED A CARTE TELL	N/N
	I NEED A CARTE L	N/N

Figure 3.1: Four examples where MaxEnt predicted labels are almost perfectly matched by the human labels.

Transcript	Hypotheses	human/maxEnt labels
what did you say	WHAT DID YOU SAY	B/N
	WHAT DID YOU SAY HERE	B/N
	WHAT DO YOU SAY	B/N
	WHAT DID YOU JUST SAY	B/N
[]	THERE	N/N
	NO	N/N
	I	N/N
	NO I	N/B
	ER I	N/N
the art house hotel	YEAH CARTE HOTEL	N/B
	THERE ARE CARTE HOTEL	N/N
	NO CARTE HOTEL	N/N

Figure 3.2: Four examples where the MaxEnt predicted wrong labels in comparison to the human labels.

The group of false-B (i.e. a hypothesis labeled with 'B' that should be labeled 'N') contains both hypotheses that should be accepted (contains at least one lower ranked hypothesis that is hand labeled with 'B') and hypotheses that should have been rejected (all hypotheses should have been labeled 'N'). The false-B were split into two classes *bad* errors and *not so bad* errors. A false positive was judged to be *bad* if it contained words that was not in the transcript but was a valid dialogue move or a better hypotheses could be found lower down the n-best list, e.g. the top hypothesis was '*yeah restaurant*' but the transcript and a lower ranked hypothesis was '*indian restaurant*'. A false positive was judged *not so bad* if no better hypothesis was in the list or it was not a valid dialogue

move in that context, e.g. answering 'yes' to the question *How can I help you?*'. In total 43 errors were classified as *not so bad* and 30 were classified as *bad*.

The group of false-N was 19 in total (i.e. n-best lists that contain a hand-labeled 'B' but MaxEnt predicts that all are 'N'). 12 of these had a 'B' labeled top-hypothesis and for the remaining 7 the 'B' was lower ranked. Three categories of prediction mistakes was identified: 3 cases of 'crosstalk' where a hypothesis was mistakenly hand labeled with a 'B', 4 cases of data sparsity (e.g. *mexican* only occurred in this set of dialogues) and for the remaining 12 the cause is unclear, but the most probable cause is some kind of 'unfortunate' combination of features. In the group of unclear cases there are also instances that usually seems to be predicted correctly like: numbers after the question *are you interested in any hotel option number?* or *no* after an appropriate question.

CHAPTER 4

Discussion and Conclusions

The remainder of this thesis will discuss the result and specific approach of the study and present which conclusions are drawn. The results from this study are limited by the performance of the speech recogniser. The oracle scores show that approximately 27% of the n-best lists does not contain an appropriate hypothesis, and although no exact figures are available it seems unlikely that the users failed to give an appropriate answer to such an extent. From the author's experience of transcribing the user 'utterances' saved in sound files it seems more likely that there are two problems with improving speech recognition for the TownInfo dialogue system that this study has not addressed:

- Mis-timed recognition - where the microphone was not activated at the right time or was deactivated before the user finished speaking.
- Bad recognition hypotheses - Where the user clearly said something other than the recogniser decoded. This problem relates to specific configurations and language model used for decoding but also to the attempt to do speaker independent recognition with both native and non-native speakers.

Assuring that as many recognition n-best lists of user utterances contain a hypothesis good enough to be dealt with by a dialogue system is absolutely crucial for a positive impact from using a context dependent statistical model such as the one described in

this study where a good hypothesis can be found only provided that it exists in the first place.

4.1 Labeling

The qualitative evaluation showed that there seems to be quite high number of mislabeled hypotheses and it can not be excluded that more consistent labeling might have improved the result slightly, but probably not to an extent that would have affected the general conclusions drawn from this study.

The labeling used in this study is closely related to the idea of concept accuracy. As described in section 2.3 the notion of concept is not uniquely defined from the dialogue system's perspective in this study, but rather from a combination of the system's and author's perspective. The labeling therefore reflect an idealised concept of how the system *should* react when given a hypothesis. The idealised concept labels are okay to use in a study such as the one in this thesis to give more insights to possibilities and limitations of the method. The idealised concept labels could even be extended to try and find e.g. the most grammatical (according to one or more parsers) of the correct hypothesis. But to train statistical models for actual use in a dialogue system the appropriate labels would be to label exactly according to the semantics of the system, although it is probably a good idea to try and learn to reject utterances that consists of crosstalk or are empty due to mis-timed recognition.

4.1.1 Automatic Labeling

Hypotheses labeled automatically and used for training does not reach the baseline when tested on human labels. They perform only slightly worse in terms of recall compared to training on human labels, but substantially worse in terms of precision compared to the best result for training on human labels. One possible reason why the automatic labels have a lower precision is that too many hypotheses that would have been labeled 'N' by a human labeler gets a 'B' label, e.g. n-best lists where an important keyword has been substituted or deleted in all or some of the hypotheses (as is the case

when figures 2.4 and 2.3 are compared). The hypothesis is confirmed by the fact that the simple label system which have less 'B' labeled hypotheses gets a 3% higher precision than the advanced label system. The most useful way to use the automatic labels at the moment would be as a set of raw labels where a human could go over the labels and just change 'bad' 'B' labels to 'N' to reduce the human labeler's cognitive load of following a large set of labeling guidelines.

A different automatic labeling system would be to use the dialogue system itself to find out which hypotheses represent the same dialogue move as the corresponding transcript, and tag those hypothesis with 'B' and the rest with 'N'. This type of labeling system would also make it easy to collect and relabel data when system behaviour is changed or developed to capture more fine grained semantic distinctions.

4.2 Features

The features were divided into three groups: discourse, acoustic and hypothesis. Most of the features have been used in previous similar studies with good results which was confirmed in this study. Most of the features used in this thesis represented properties of the recognition hypothesis and the only feature that wasn't related to the immediate context of the current task was the *previous asrInput*, all other features like e.g. previous system question, current task or recognition hypothesis, are all from the immediate context of the utterance. Since only *previous asrInput* are represented from dialogue discourse not in immediate context with the utterance the question of how much and how far back previous context can influence dialogue relevance remains to some extent unanswered, except that immediate context features seem almost enough, at least for this domain.

Previous asrInput was a feature that to my knowledge never been used in a similar task before and although it improved the result for the small feature set it was opaque in terms of qualitative impact, but since it does have a positive impact it is worth to investigate further. A first step would be to limit it to a reasonable number instead of *all* or limit it to the utterances in the current overall task, e.g. booking a hotel room.

Features that were 'on the list' to try, because of positive impact on the result in other studies, but wasn't tried because of lack of time were more acoustic features, in particular the RMS acoustic measures, and recognition confidence scores from the hypotheses, both for whole hypothesis and single words. It is also worth trying some more different feature representations of current features, e.g. setting discrete threshold values of bigram frequency, duration, min amplitude based on more exact data distribution measures. Another set of features that would be interesting to try is to align each hypothesis in the n-best list with the dialogue move (or speech act, task, slot value or all of them) that they *would* be aligned with if used as system input, to use both for training and test. In real use it might slow down the system, but if done well it would only marginally affect the flow of the dialogue.

The linguistic features; words and full hypotheses, are strong features when predicting which hypotheses are 'B' and 'N' labeled. This was supported by other studies and intuitively it is no surprise, especially not for a task driven dialogue, since it seems plausible that the semantic interpretation of the question is reflected in a valid answer. For example there are only four types of valid answers to the dialogue system's question *would you like something cheap, expensive, or reasonably priced?*, three which are obvious from the alternatives in the question and the fourth being neutral, a fifth alternative is to not acknowledge it as a valid question if the system has misunderstood the state of the dialogue. The five types of answer *can* be realised in a 'million' ways but there are only a few realisations that are commonly used and that the system can deal with, so it seems reasonable that a statistical model reflects this. A question such as *how can i help you?* might seem more open, but in the context of a task driven dialogue system there are again a limited number of tasks and common patterns in the way user express his/her answer.

The general impression from most of the features except the linguistic features is that they provide only a small gain and that the substantial gain lies in combinations of features. The combination of features seemed to interact in an intricate way so that adding a feature to a feature set at one point increased error, but adding the same feature at a

later point reduced error. A hypothesis why the big feature set was better than the small feature set might be that it combined features from several different sources; acoustic, linguistic, dialogue discourse, parsability and n-best list features that helped in creating a more robust model.

4.3 Method

The Maximum Entropy modeling method and specific toolkit seems to be well suited for modeling context dependent speech recognition. The method doesn't give as easy accessible feedback regarding which features was important as e.g. RIPPER's 'if-then' rules, and the only insight to important features comes from the building of the feature sets.

The feature sets was built up with 'trial and error' as systematically as possible, but a better way to find good feature sets would be to automatise the procedure: implement a set of possible features and set up a software environment to try different feature sets and number of iterations. To minimise what might become a combinatory explosion a 'trial and error' method like the one in this study could be used to constrain the possible combinations, e.g. previous system question and hypothesis could always be present in the set.

False positives are a natural part of communication and is probably the most obvious drawback of context dependent interpretations. But they should be avoided within reason, not at all price, the important thing is that the dialogue system can handle false positives in a good way. A drawback from using linguistic features (words and hypotheses) might be that the model created is very rigid and in effect behaves like a recognition grammar, i.e. nothing else can be 'recognised'. This shouldn't be a problem provided that the training data covers a reasonable extent of possible phenomena that the dialogue system can handle.

4.3.1 Calculation Problems

During the qualitative evaluation in the very end of the project it was noted that the script used for calculating number of *correct-B* and *false-rejects* produced the wrong numbers. The script for calculating *correct-B* was fixed and as a result seriously impaired what until then was the believed result, the result section contain these new 'worse' but correct results. A failed attempt was made to fix also the calculation of *false rejects*. The script underestimates the true number of *false rejects* and since *false-B* was calculated through subtractions of other categories from the total they are overestimated. This means that the precision figures are in reality slightly higher for predicting 'B' labels, and precision figures for *rejects* are slightly lower. All other calculations in the result are correct.

4.4 Results

Neither the small nor big feature set beat the baseline in terms of correctly predicting 'B' labeled hypotheses, i.e. hypotheses that the dialogue system should accept. The big feature set performed clearly better than the small feature set getting a 5 percentage units higher precision and approximately 2 percentage units higher recall. The big feature set beat the baseline in 5 out of the 10 test batches, but overall didn't reach baseline performance in retrieving 'B' labeled hypotheses.

The big feature set was also better than the small feature set at correctly rejecting n-bests lists that don't contain a good hypothesis, in particular it had a much higher (11%) recall. Since there was no baseline to compare the reject scores with it is hard to say whether they are 'good enough' or not. Intuitively the score seems promising in terms of further development but they are not yet recommended for actual use in a dialogue situation due to too low precision that would cause false rejections of valid hypotheses.

In the reranking interpretation of the 'B' and 'N' labels where a 'B' label outranks an 'N' label the intention was similar to the accept and reject interpretation in that the predicted labels should be the same as the hand labels. But if the labels are used as pure reranking markers then the false rejects where the top hypothesis has a hand labeled 'B' will

also be used by the dialogue system resulting in retrieving 94.9% (big feature set) and 94.1% (small feature set) of the oracle score which in effect means a substantial relative reduction of error; 44.5% and 35.9% respectively. The remaining good hypotheses being incorrectly ranked due to false rejects and false positives where a better hypothesis was ranked lower than top.

Both the reranking and accept/reject approach also results in several false positives. In the qualitative evaluation (section 3.4) the false positives was divided into '*bad*' and *not so bad* based on where they occurred. No comparison was made whether this would be more or less than when the top-hypothesis was used, but the general impression was that the false positives increased due to hypotheses containing keywords tended to get labeled with a '*B*'. However the qualitative evaluation was perhaps too harsh in classifying predictions as *bad*, e.g. when the top hypothesis was '*yeah restaurant*' but the transcript and a lower ranked hypothesis was '*indian restaurant*'. It is clearly better in this context to choose '*yeah restaurant*' that is at least half right instead of e.g. '*irish pub*' that is completely wrong.

4.5 Future Work

Some ideas of future work have already been mentioned in previous paragraphs in this chapter: better (automatic) labeling, automatic search for good feature sets and more possible features. A more thorough qualitative analysis is also necessary to be able to search for good features with an intended effect in mind.

An important future work would also be to improve the performance, in particular precision of *accepts* and *rejects*, because if a highly predictable subset of utterances can be spotted then the dialogue manager can use this subset and adapt its strategy accordingly.

The next step in this work would be to integrate a trained MaxEnt model into the OOA structure of the TownInfo dialogue system and evaluate the performance, in particular with regards to the false positives. Integrating the model in the dialogue system would

presumably change the dialogue flow and depending on which features are used the model might behave differently due to changed contextual environment. A recommendation would therefore be to collect new data and retrain the model.

4.6 Conclusions

The big feature set was better than the small feature set on both the *accept* and *reject* task but neither beat the baseline.

- **The Small Feature Set:** hypothesis, previous system question, every word in hypothesis, all previous asrInput.
- **The Big Feature Set:** hypothesis, previous system question, min amplitude, number of hypothesis, parsability, current task, hypothesis rank, length of hypothesis, bigram + relative freq.

If the *accept* and *reject* perspective is adopted and integrated it is best to treat them as *recommendations* to the dialogue manager rather than definite decisions. The recommended strategy for an immediate future is to use the reranking approach in the dialogue system thus retrieving 94.9% of the oracle score and a relative reduction of error of 44.5%.

This thesis has showed that building context based statistical models for reranking recognition n-best lists can improve speech recognition for a task driven dialogue system. It has also shown that close context features - related to the current task provides the most important contextual cues. Integration of contextual information has thus strengthen the common sense intuition that a valid interpretation of an utterance is context dependent.

APPENDIX A

Example of Results

Explanation to the structure of the appendix:

Dialogue name: incar_logThuMar216:32:182006.txt

System's question: how can i help you?

Transcript: TransInput: indian restaurant

HYPOTHESIS-HUMAN-LABEL MAXENT-LABEL:

YEAH RESTAURANT-N B

IT YEAH RESTAURANT-N N

INDIAN RESTAURANT-B B

ANY RESTAURANT-N N

IN YEAH RESTAURANT-N N

YEAH RESTAURANTS-N B

INDIAN RESTAURANTS-B B

Features used to train and test, the big feature set: hypothesis, system out, length, task, min amp, no of hypos, parsability, rank, bigram + freq.

incar_logThuMar216:32:182006.txt

how can i help you?

TransInput: indian restaurant

YEAH RESTAURANT-N B

IT YEAH RESTAURANT-N N

INDIAN RESTAURANT-B B

ANY RESTAURANT-N N

IN YEAH RESTAURANT-N N

YEAH RESTAURANTS-N B

INDIAN RESTAURANTS-B B

did you say indian restaurant?

TransInput: yes

YES-B B

what kind of food would you like?

TransInput: indian

IN YEAH-N N

IT YEAH-N N

INDIAN-B B

A IN YEAH-N N

IN YEAH THE-N N

IT YEAH THE-N N

THE A IN YEAH-N N

would you like something cheap, expensive,
or reasonably priced?

TransInput: reasonably priced

REASONABLY_PRICED-B B

REASONABLY_PRICED FOOD-B B

do you want something central?

TransInput: yes

YES-B B

are you happy with that choice?

TransInput: yes

YES-B B

YES I-B B

YES ER-B B

would you like to search some more?

TransInput: no

NO-B B

NO ALL-B B

NO I-B B

NO ALL I-B B

NO OR-B B

NO I I-B B

NO ALL I I-B B

incar_logThuMar216:35:022006.txt

how can i help you?

TransInput: find a bar

I'D LIKE TO-N N

I'D LIKE-N N

I WOULD LIKE TO-N N

I WOULD LIKE-N N

I'D LIKE A-N N

I DON'T LIKE TO-N N

I DON'T LIKE-N N

how can i help you?

TransInput: bar

BYE-N N

A BAR-B N

BAR-B N

AH-N N

BY-N N

A BAR AH-B N

BAR AH-B N

how can i help you?

TransInput: pub

A PUB-B B

PUB-B B

how can i help you?

TransInput: id like to go out for a drink

I'D LIKE TO GO FOR A DRINK-B B

I'D LIKE A TOUR FOR A DRINK-N N

I'D LIKE TO GO FOR DRINK-B B

I'D LIKE A TOUR FOR DRINK-N N

I'D LIKE TO GO OR FOR A DRINK-N B

I'D LIKE A TOUR AT FOR A DRINK-N N

I'D LIKE A GO FOR A DRINK-B B

did you say drink?

TransInput: yes

YES-B B

YES UM-B B

YES I-B B

what type of bar are you looking for?

Maybe a jazz bar or a cafe?

TransInput: cafe

A CAFE-B B

THE CAFE-B B

CAFE-B B

THE A CAFE-B B

IT CAFE-B B

THE THE CAFE-B B

would you like something cheap, expensive,
or reasonably priced?

TransInput: reasonably or expensive

THREE DOUBLE YOU OR EXPENSIVE-B N

THREE DOUBLE A OR EXPENSIVE-B N

THREE DOUBLE YOU OR A EXPENSIVE-B B

THREE DOUBLE A OR A EXPENSIVE-B N

THREE DOUBLE YOU OR IT EXPENSIVE-B B

THREE DOUBLE YOU OR INEXPENSIVE-N N

THREE DOUBLE A OR IT EXPENSIVE-B N

do you want something central?

TransInput: yes

YES-B B

YES I-B B

would you like to search some more?

TransInput: yes find me a bar thats further out

YES FOR ME A BAR THAT'S FURTHER AT-N B

YES BY ME A BAR THAT'S FURTHER AT-N B

YES FOR ME A BAR THAT'S FURTHER-N B

YES FOR ME A BAR THAT'S FURTHER OUT-B B
YES BY THE A BAR THAT'S FURTHER AT-N B
YES BY ME A BAR THAT'S FURTHER-N B
YES FOR ME A BAR THAT'S FURTHER TO-N B

how can i help you?

TransInput: drink

THANK YOU-N N

DON'T CARE-N N

I DON'T CARE-N B

DON'T-N N

THANKS-N N

CAN RING YOU-N N

I DON'T-N N

how can i help you?

TransInput: like to go out for drinks

I WILL FOR DRINKS-B N

I DON'T OR FOR DRINKS-N N

WHERE DO OR FOR DRINKS-B N

WILL YOU WILL FOR DRINKS-B N

I WOULD YOU WILL FOR DRINKS-B N

WOULD YOU WILL FOR DRINKS-B N

WHERE DO WILL FOR DRINKS-B N

how can i help you?

TransInput: cafe

CAFE-B B

A CAFE-B B

CAFE OR-B B

A CAFE OR-B B

did you say cafe ?

TransInput: yes

YES-B B

what type of bar are you looking for?

Maybe a jazz bar or a cafe?

TransInput: cafe

OKAY-N N

OK-N N

ARE THERE A-N N

A-N N

ER FOR A-N N

AH FOR A-N N

I FOR A-N N

would you like something cheap, expensive,
or reasonably priced?

TransInput: mid range or expensive

NEAR RANGE OR EXPENSIVE-B N

NEAR THE RANGE OR EXPENSIVE-B N

NO THE RANGE OR EXPENSIVE-B N

NEAR RANGE OR A EXPENSIVE-B B

NEAR THE RANGE OR A EXPENSIVE-B B

NO IN THE RANGE OR EXPENSIVE-B N

NEAR RANGE OR OR EXPENSIVE-B N

do you want something central?

TransInput: no

NO ALL-B B

NO-B B

NO I'LL-B B

NO ALL I-B N

NORMAL-N N

NO I-B B

would you like to search some more?

TransInput: yes

YES-B B

YES I-B B

THE YES-B B

YES ON-B B

THE YES I-B B

THE YES ON-B B

how can i help you?

TransInput: cafe

CAFE-B B

CAFE OR-B B

did you say cafe ?

TransInput: yes

YES-B B

YES I-B B

what type of bar are you looking for?

Maybe a jazz bar or a cafe?

TransInput: cafe

CAFE-B B

A CAFE-B B

CAFE OR-B B

A CAFE OR-B B

would you like something cheap, expensive,
or reasonably priced?

TransInput: i dont care

I DON'T CARE-B B

DON'T CARE-B N

AH I DON'T CARE-B B

I I DON'T CARE-B B

ER I DON'T CARE-B B

ER I I DON'T CARE-B B

I I I DON'T CARE-B B

do you want something central?

TransInput: yes

YES-B B

would you like to search some more?

TransInput: okay

OKAY-B B

OK-B B

OK ER-B B

THE OK-B B

THE OK ER-B N

how can i help you?

TransInput: find me bar

I NEED A BAR-B B

I MEAN A BAR-B N
CARTE ME A BAR-N B
AH I NEED A BAR-B B
I MEAN BAR-B N
I I NEED A BAR-B B
I'LL NEED A BAR-B N

did you say bar?

TransInput: yes

YES-B B

what type of bar are you looking for?

Maybe a jazz bar or a cafe?

TransInput: i dont mind

I DON'T MIND-B B
NO I DON'T MIND-N N
ER I DON'T MIND-B N
I I DON'T MIND-B B
AH I DON'T MIND-B B
I DON'T MIND UH-B B
NO I I DON'T MIND-N N

would you like something cheap, expensive,
or reasonably priced?

TransInput: whatever

WHATEVER-B B
WHAT THERE-N N
WHAT OTHER-N N

do you want something central?

TransInput: i dont care

I DON'T CARE-B B

I I DON'T CARE-B B

AH I DON'T CARE-B B

UH I DON'T CARE-B B

HOW I DON'T CARE-B B

AH I I DON'T CARE-B B

I I I DON'T CARE-B B

are you interested in any bar option number

or would you like me to

keep going?.

TransInput: keep going

KEEP GOING OR-B B

KEEP GOING ON-B

B KEEP GOING-B B

KEEP GOING OR UM-B B

KEEP GOING ON A-B B

KEEP GOING OR A-B B

KEEP GOING OR ON-B B

are you happy with that choice?

TransInput: no

NO-B B

NO HELLO-B B

NO ALL-B B

NO HOW-B B

NO I-B B

NO ALL I-B B

NO I I-B B

how can i help you?

TransInput: cant help me

DON'T KNOW ME-N N

DON'T HAVE ME-N N

DON'T HAVE A-N N

I DON'T KNOW ME-N N

I DON'T HAVE ME-N N

I DON'T HAVE A-N N

DON'T HAVE ANY-N N

how can i help you?

TransInput: [crosstalk]

WHAT WHAT NUMBER-N N

WHAT WHAT NUMBER AH-N N

WHAT WHAT IT AND THERE-N N

WHAT WHAT AND THERE-N N

WELL WHAT NUMBER-N N

WHAT WHAT IT NUMBER-N N

WHAT WHAT ARE THE NUMBER-N N

incar_logThuMar216:42:412006.txt

how can i help you?

TransInput: double room at the hotel royal

A DOUBLE ROOM AT THAT HOTEL RIGHT NOW-B B

DOUBLE ROOM AT THAT HOTEL RIGHT NOW-B B

A DOUBLE ROOM AT THAT HOTEL RIGHT HOUR-B B

DOUBLE ROOM AT THAT HOTEL RIGHT HOUR-B B

A DOUBLE ROOM AT THAT HOTEL RIGHT YOU ARE-B B

DOUBLE ROOM AT THAT HOTEL RIGHT YOU ARE-B B

A DOUBLE ROOM AT THAT HOTEL RIGHT ELSE-B B

did you say double room?

TransInput: yes

YES-B B

do you want something central?

TransInput: yes

YES-B B

YES I-B B

do you need a single or double room?

TransInput: double

DOUBLE-B B

A DOUBLE-B B

DOUBLE ROOM-B B

A DOUBLE ROOM-B B

would you like something cheap, expensive,
or reasonably priced?

TransInput: expensive

EXPENSIVE-B B

A EXPENSIVE-B B

are you interested in any hotel option
number or would you like me to
keep going?.

TransInput: yes

YES-B B

ER YES-B B

YES I-B B

ER YES I-B B

are you interested in any hotel option
number or would you like me to
keep going?.

TransInput: number two

NUMBER TWO-B B

NUMBER TO-N N

NUMBER TOO-N N

NUMBER AT TWO-B B

NUMBER A TWO-B B

NOT A TWO-N N

NOT AT TWO-N B

are you happy with that choice?

TransInput: yes

YES-B B

YES UM-B B

would you like to search some more?

TransInput: yes

YES-B B

YES I-B B

how can i help you?

TransInput: id like to find a restaurant

I'D LIKE TO FIND A RESTAURANT-B B

I'D LIKE TO FIND RESTAURANT-B B

I'D LIKE TO PLAN A RESTAURANT-N B

I'D LIKE A FIND A RESTAURANT-B B
 I LIKE TO FIND A RESTAURANT-B B
 I'D LIKE TO FIND A RESTAURANT AT-B B
 I'D LIKE A TO FIND A RESTAURANT-B B

did you say restaurant?

TransInput: yes

YES-B B

what kind of food would you like?

TransInput: i dont care

I DON'T CARE-B B

CAN I DON'T CARE-B N

I I DON'T CARE-B B

DO I DON'T CARE-B N

AH I DON'T CARE-B B

CAN I I DON'T CARE-B N

DO I I DON'T CARE-B N

would you like something cheap, expensive,
 or reasonably priced?

TransInput: something expensive

SOMETHING EXPENSIVE-B B

SOMETHING INEXPENSIVE-N N

TELL ME EXPENSIVE-N B

SOMETHING A EXPENSIVE-B B

SOMETHING YOU EXPENSIVE-B N

TELL ME A EXPENSIVE-N N

SOMETHING A A EXPENSIVE-B B

do you want something central?

TransInput: i dont care

I DON'T CARE-B B

I I DON'T CARE-B B

AH I DON'T CARE-B B

AH I I DON'T CARE-B B

I I I DON'T CARE-B B

which restaurant option number would you
like to choose?.

TransInput: number two

NUMBER TWO-B B

NOT A TWO-N N

NUMBER TO-N N

NUMBER DO-N N

NO I'M AT TWO-B B

NONE OF TO-N N

NOT A TO-N N

are you happy with that choice?

TransInput: yes

YES-B B

YES ER-B B

would you like to search some more?

TransInput: no thanks

NOT BAD IT-N N

NOT A IT-N N

NOT A TO-N N

NOT I TO-N N

NOT VERY TO-N N

NOT THERE IT-N N

NOT I'D TO-N N

how can i help you?

TransInput: no help please

NOT NOT PLEASE-N N

NOT CARTE PLEASE-N N

NOT PLEASE-N N

NOT HOW PLEASE-N N

NOT OR PLEASE-N N

NOT ARE PLEASE-N N

A NOT NOT PLEASE-N N

did you say pub?

TransInput: no goodbye

NO GOODBYE-B B

NO NO GOODBYE-B B

THE NO GOODBYE-B B

NO GOODBYE THE-B B

NO GOODBYE OF-B B

NO GOODBYE OF THE-B B

NO GOODBYE AND THE-B B

incar_logTueFeb2813:19:052006.txt

how can i help you?

TransInput: im looking for a hotel

I'M LOOKING FOR A HOTEL-B B

I'M LOOKING FOR HOTEL-B B

I'M LOOKING FOR THE HOTEL-B B
I'M LOOKING FOR AN HOTEL-B B
I AM LOOKING FOR A HOTEL-B B
I'M LOOKING FOR THE TOWER-N N
I'M LOOKING FOR ER A HOTEL-B B

did you say hotel?

TransInput: yes yes

YES-B B

YES I-B B

do you want something central?

TransInput: yes

YES-B B

YES MM-B B

YES I CAN-B B

YES THE-B B

YES AND-B B

THE YES-B B

YES CAN-B N

do you need a single or double room?

TransInput: i dont mind

I DON'T MIND-B B

I I DON'T MIND-B B

AH I DON'T MIND-B B

I'M DON'T MIND-B B

would you like something cheap, expensive,
or reasonably priced?

TransInput: something reasonable

SOMETHING REASONABLE-B B

which hotel option number would you like
to choose?.

TransInput: number two

NUMBER TWO-B B

NUMBER TO-N N

A NUMBER TWO-B B

NUMBER TOO-N B

NUMBER OF TO-N N

NUMBER A TWO-B B

NUMBER ER TWO-B N

are you happy with that choice?

TransInput: yes

YES-B B

YES IN-B B

YES I-B B

YES AND-B B

THE YES-B B

THE YES IN-B B

THE YES I-B B

would you like to search some more?

TransInput: no thanks

NO THANKS-B B

NO THANKS I WOULD-B B

References

- Berger, A., Della Pietra, S. and V., D. P. (1996), 'A maximum entropy approach to natural language processing', *Computational linguistics* **22**(1), 39–71.
- Boros, M., Eckert, W., Gallwitz, F., Görz, G., Hanrieder, G. and Niemann, H. (1996), Towards understanding speech: Word accuracy vs. concept accuracy, in 'Proceedings of ICSLP-96', Philadelphia, USA, pp. 1005–1008.
- Bos, J., Klein, E., Lemon, O. and Oka, T. (2003), Dipper: Description and formalisation of an information-state update dialogue system architecture, in '4th SIGdial Workshop on Discourse and Dialogue', Sapporo, Japan, pp. 115–124.
- Cheyen, A. and Martin, D. (2001), 'The open agent architecture', *Journal of Autonomous Agents and Multi-Agent Systems* **4**(1), 143–148.
- Chotimongkol, A. and Rudnicky, A. (2001), N-best speech hypotheses reordering using linear regression, in 'Proceedings of EuroSpeech', pp. 1829–1832.
- Cohen, W. (1996), Learning trees and rules with set-valued features, in 'AAAI'.
- Daelemans, W., Zavrel, J., van der Sloot, K. and van den Bosch, A. (2002), *TiMBL: Tilburg Memory Based Learner*, 4.2 edn. Reference Guide, ILK Technical Report 02-01.
- Gabsdil, M. and Lemon, O. (2004), Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems, in 'Proceedings of ACL', Barcelona, Spain, pp. 343–350.
- Holmes, J. and Holmes, W. (2001), *Speech Synthesis and Recognition*, Taylor & Francis, London, England.

- Jonsson, R. (2006), Dialogue context-based re-ranking of asr hypothesis. (submitted to SLT).
- Le, Z. (2004), *Maximum Entropy Modeling Toolkit for Python and C++*. Technical Manual.
- Lemon, O., Georgila, K. and Henderson, J. (2006), Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: The talk towninfo evaluation. submitted to SLT.
- Lemon, O., Georgila, K., Henderson, J. and Stuttle, M. (2006), An isu dialogue system exhibiting reinforcement learning of dialogue policies: Generic slot-filling in the talk in-car system, *in* 'EACL', Trento, Italy, pp. 119–122.
- Litman, D., Hirschberg, J. and Swerts, M. (2000), Predicting automatic speech recognition performance using prosodic cues, *in* 'Proceedings of NAACL-00'.
- Litman, D. and Pan, S. (2000), Predicting and adapting to poor speech recognition in a spoken dialogue system, *in* 'Proceedings of AAAI', Austin, USA, pp. 722–728.
- Malouf, R. (2002), A comparison of algorithms for maximum entropy parameter estimation, *in* 'Proceedings of CoNLL-2002', pp. 49–55.
- Meza-Ruiz, I. and Lemon, O. (2005), 'Using dialogue context to improve parsing performance in dialogue systems', International Workshop on Computational Semantics (IWCS). Tilburg.
- Och, F. and Ney, H. (2002), Discriminative training and maximum entropy models for statistical machine translation, *in* 'Proceedings of the 40th Annual Meeting of ACL', Philadelphia, USA, pp. 295–302.
- Purver, M., Ratiu, F. and Cavedon, L. (2006), Robust interpretation in dialogue by combining confidence scores with contextual features, *in* 'ICSLP', Pittsburgh, USA. *to appear*.
- Ratnaparkhi, A. (1996), A maximum entropy model for part-of-speech tagging, *in* 'Proceedings of EMNLP'.
- Tjong Kim Sang, E. and De Meulder, F. (2003), Introduction to conll-2003 shared task: Language independent named entity recognition, *in* 'Proceedings of CoNLL-2003', Edmonton, Canada, pp. 142–147.

- Walker, M., Wright, J. and Langkilde, I. (2000), Using natural language processing and discourse features to identify understanding errors in a spoken dialogue system, *in* 'Proceedings of ICML-2000'.
- Weilhammer, K., Stuttle, M. and Young, S. (2006), Bootstrapping language models for dialogue systems, *in* 'Proceedings of ICSLP'. *to appear*.
- Young, S. (2004), *ATK An Application Toolkit for HTK*, 1.4.1 edn. Technical Manual.
- Zhai, L., Fung, P., Schwartz, R., Carpuat, M. and Wu, D. (2004), Using n-best lists for named entity recognition from chinese speech, *in* 'Proceedings of NAACL'. (short papers).